AN ALGORITHM FOR COMPUTING OPTIMUM STOCK LEVELS IN A TWO-LEVEL MAINTENANCE SYSTEM

TECHNICAL REPORT

JANUARY 1970

By
W. KARL KRUSE

UNITED STATES ARMY LOGISTICS MANAGEMENT CENTER
Fort Lee, Virginia
AN ALGORITHM FOR COMPUTING OPTIMUM STOCK LEVELS
IN A TWO LEVEL MAINTENANCE SYSTEM

TECHNICAL REPORT
JANUARY 1970

By:
W. Karl Kruse
AMC INVENTORY RESEARCH OFFICE

UNITED STATES ARMY LOGISTICS MANAGEMENT CENTER
FORT LEE, VIRGINIA
ABSTRACT

An optimization algorithm for a multi-echelon model is given which does not assume convexity. The algorithm employs a bound technique to determine a finite search area. Updating the bound is used to increase the efficiency.
An Algorithm for Computing Optimum Stock Levels in a Two Level Maintenance System

Introduction

In general, inventory models are optimized by some appropriate method which depends upon convexity of the objective function. Thus, differentiation and difference equation techniques are used to find the conditions for optimality. Convexity, however, frequently is not seen where there is more than one decision variable. Nevertheless, optimization assuming convexity will usually produce nearly optimal values due to the general flat nature of inventory model equations in the area of the optimum. With this in mind, an algorithm which assumed convexity for minimizing the total cost in a two level aircraft maintenance system was used. During observation of the algorithm it was noticed that small changes in critical parameters sometimes caused large changes in the optimal stock distribution. Even though these changes actually resulted in relatively small changes to total cost, the inconsistency of the algorithm was undesirable in view of the time phased nature of the supply control study in which it would be employed, for over the time period some or all of the parameters will change. This paper presents an algorithm developed to find the exact minimal cost distribution of assets and eliminate the inconsistencies.

For a thorough discussion of the model see reference 1. An almost identical model developed independently at about the same time as the model of reference 1 is discussed in 2. A sufficient description of the model follows for understanding of the optimization problem.

Both models are concerned with optimizing stock levels of high cost -
low demand repairable items in a two echelon maintenance system. There is
a single top echelon maintenance facility which does overhaul and complex
repair. The lower echelon consists of several units which are capable of
doing non complex repair. Demands for items are placed upon the lower
echelons. When an item is demanded it is accompanied by the return of a
failed item. The item can be repaired at the lower echelon with a probability
f and it can be repaired at the top echelon with probability 1-f.

For this class of items the optimal policy at lower and top echelons
is of the form (S,S-1). Demands are assumed to be compound Poisson at each
lower echelon unit. Under these assumptions an analytic formulation is
possible.

The objective function to be minimized is

\[ TC = \sum_{i=0}^{N} C_h S_i + \sum_{i=1}^{N} C_b B_i(S_i, \lambda_i, T_{ir}, T_{is}, W_o) \]

where

- \( S_o \) = top echelon spares
- \( S_i \) = spares at unit i of lower echelon
- \( \lambda_i \) = average failures per day at unit i
- \( f_i \) = percent of total failures repaired at unit i
- \( T_{ir} \) = repair time at i\textsuperscript{th} unit
- \( T_{is} \) = replenishment time for i\textsuperscript{th} unit
- \( W_o \) = average wait for stock at top echelon
- \( B_i \) = average backorders at unit i
- \( C_h \) = holding cost per spare per year
- \( C_b \) = backorder cost per spare per year
TWO LEVEL MAINTENANCE SYSTEM
Notice that

\[ \text{TC}(S_0; S) = C_h S_0 + \sum_{i=1}^{N} \left[ C_{h_i} S_i + C_{b_i} B_i(S_i \lambda_i f_i T_i S_i W) \right] \]

or

\[ \text{TC}(S_0; S) = C_h S_0 + \text{TCA}(S_0; S) \]

where \( \text{TCA}(S_0; S) \) is total lower echelon cost given a top echelon stock of \( S_0 \), and lower level stock vector \( S = (S_1, S_2, \ldots, S_n) \).

Any given \( S_0 \) establishes the value of the parameters of the inventory level probability distribution which are used to compute \( B_i(\cdot) \). With parameters fixed, \( B_i(\cdot) \) is a convex function over \( S_i \). Thus, minimization of \( \text{TCA}(S_0; S) \) can easily be done. This is the basic computational feature on which the algorithm rests.

**Mathematical Basis**

Before presenting the algorithm the following theorems need to be proved.

**Theorem 1.** If \( S^*_o \) is the top echelon stock at an optimum, a necessary condition for optimality is that \( \text{TCA}(S^*_o; S) \) be minimized with respect to \( S = (S_1, S_2, \ldots, S_n) \)

**Proof.** Let \( S^* \) be the vector of lower echelon stock levels which minimize \( \text{TCA}(S^*_o; S) \) and \( S' \) be any other stock level vector. Now

\[ \text{TC}^* = C_h S^*_o + \text{TCA}(S^*_o; S^*) < C_h S^*_o + \text{TCA}(S^*_o; S') \]

Thus, \( S' \) cannot minimize total cost.
Theorem 2. Let $S^*(j)$ denote the vector of optimum lower echelon stock levels when $S_o$ is $j$. Then

$$S^*(k) \leq S^*(j) \text{ if } j < k$$

Proof. Since each $S_i$ is determined independently it is sufficient to show that any $S^*(k) \leq S^*(j)$, $i = 1,2, \ldots N$

When $j < k$ the mean of the compound Poisson distribution of items in the pipeline is greater for $S_o = j$ than for $S_o = k$. Let $\mu(j)$ be the mean for $S_o = j$. Then $\mu(j) > \mu(k)$. The condition for optimality at a lower echelon unit is

$$P(S+1/\mu) \leq \frac{C_h}{C_b} \leq P(S/\mu)$$

where

$$PS/\mu = \sum_{j=S}^{\infty} P(j/\mu)$$

and $p(j/\mu)$ is the compound Poisson probability. Now $P(S/\mu(k)) < P(S/\mu(j))$ and the optimality condition cannot be satisfied with a larger $S$ if $\mu(j) > \mu(k)$.

Theorem 3. Let $TCA(j,S^*(j))$ denote optimum lower echelon cost when $S_o = j$

Then $TCA(j,S^*(j)) > TCA(k,S^*(k))$ if $j < k$

Proof. Let $S^*(j)$ be vector of optimum lower echelon stock levels when $S_o$ is $j$.

Now $TCA(j,S^*(j)) > TCA(k,S^*(j)) \geq TCA(k,S^*(k))$

The first inequality follows from the fact that backorder cost is reduced when top echelon stock is increased.
Theorem 4. Let $TC$ be any total cost and $TCA(\infty, S^*(\infty))$ be the optimum lower echelon cost when top echelon stock is infinite (eliminates $W_0$). Then an upper bound

$$b = \left\lfloor \frac{(TC - TCA(\infty, S^*(\infty)))}{C_h} \right\rfloor$$
onumber

on top echelon stock can be established such that $S^*_0 \leq b$.

Proof. $TC \geq TC^* = C_h S^*_0 + TCA(S^*_0, S^*(S^*_0)) \geq C_h S^*_0 + TCA(\infty, S^*(\infty))$

Therefore

$$\left\lfloor \frac{(TC - TCA(\infty, S^*(\infty)))}{C_h} \right\rfloor \geq S^*_0$$

where $\left\lfloor x \right\rfloor$ denotes the largest integer $\geq x$

A verbal proof of the above is that if you must spend at least $TCA(\infty, S^*(\infty))$ ($TCA(\infty, S^*(\infty))$ is a lower bound on lower echelon cost) and an allocation has been found which costs $TC$, then why spend any more on upper level inventory than $TC - TCA(\infty, S^*(\infty))$. There are two features to the computation of this bound which enable recomputation to find a possible smaller upper bound. Suppose an initial upper bound $b_1$ has been found. Then by Theorem 2 we know the least cost at the lower echelon will be $TCA(b_1, S^*(b_1))$. Thus a new bound, $b_2 = \left\lfloor \frac{TC - TCA(b_1, S^*(b_1))}{C_h} \right\rfloor$ can be computed.

Likewise, a $b_3$ can be calculated as $b_3 = \left\lfloor \frac{(TC - TCA(b_2, S^*(b_2))}{C_h} \right\rfloor$.

This continues until there is no change in the bound. Another source of improvement is from new allocations which reduce $TC$. Thus each time a lower $TC$ is found the bound is recomputed.

The measure of efficiency for any optimization algorithm is the time it takes to find the optimum. In coding the algorithm for a computer we were concerned therefore with time and not necessarily the efficiency of the bound. The bound should be reduced only if the computational time in reducing the bound results in at least as large a saving in other computer time.
Quickly, it was learned that the first means described for reducing the bound cost more than it returned. The test algorithm only employed the second method. Appendix A gives a listing of the coded algorithm which was run on a time sharing system (Com-Share) in the XTRAN language. A narrative description of the algorithm follows.

The Algorithm

Step 1. Set \( S_0 = 0 \) and find \( TCA(0, S^*(0)) \) by incrementing the lower stocks independently until a minimum is reached. Set \( TCA(0, S^*(0)) \) equal to current minimum, \( CM \), and save stock distribution.

Step 2. Set upper echelon backorders to 0 and find \( TCA(\infty, S^*(\infty)) \).

Step 3. Set bound, \( b = \frac{(TCA(0, S^*(0)) - TCA(\infty, S^*))}{C_h} \).

Step 4. If \( S_0 \leq b \) go to Step 7.

Step 5. Increment \( S_0 \) by 1 (Say \( S_0 = j \)) and find \( TCA(j, S^*(j)) \). Compute \( TC(j, S^*(j)) = C_h \cdot j + TCA(j, S^*(j)) \).

Step 6. If \( TC(j, S^*(j)) < CM \), set \( CM = TC(j, S^*(j)) \), save stock distribution, recompute \( b = \frac{(TC(j, S^*(j)) - TCA(\infty, S^*(\infty))}{C_h} \), and go to Step 4.

Step 7. Output optimal stock distribution and average backorders at optimum.

Step 8. Stop.

The algorithm must pass through the minimum cost point. Whereas if the \( TC(j, S^*(j)) \) were convex an optimization procedure could stop after examining only one additional point beyond the optimum, this algorithm must
examine several until the bound is reached. One way of judging the effectiveness of the algorithm is by the number of additional points it must examine beyond the optimum before stopping. In testing of the algorithm over a rather wide range of conditions it was found that the final bound established was always close to 10% greater than the upper echelon stock at minimum. This was considered to be good.

Notice that to insure reaching the exact optimum, all possibilities of upper echelon stock up to the bound must be examined. Were the function convex, more efficient methods of converging to the optimum could be used. A substantial amount of computing speed is being lost to insure achieving an exact optimum, which in turn insures a consistent solution pattern when parameters are varied. Consistency, however, was an overriding consideration. The algorithm is being used in a supply control study to find the optimum requirements over a 5 year time span. Characteristically, some parameters will display slight changes over the 5 years. The optimum requirements solution should display a logically changing characteristic depending upon the parameter changes. If the solution were to change substantially (this can happen with little change in total cost) a good procurement schedule would be impossible. Moreover, management would have little faith in the solution if it were to change significantly without logical justification.

This version of the optimizing routine is not considered to be final. Additional research will be done to make the scheme more efficient by reducing the number of required computations.
SUBROUTINE OPT(UTL,UL,SLTL,NB,AMT,EVAC,AMDAL,COMUP,ACUP,INV,WOPT) 
REAL NSHIGH,NSLOW
C0MM0N/ID/ANAME,DATE
C0MM0N /ALLC/MM,NR,BACKO
DIMENSION AMT(NB),EVAC(NB),NR(25),NROPT(25)
ANAME=4H0PT
DATE=6H092669
NSTART=0
BAVG=UTL
NTOT=0
BTOT=0.0
DO 20 I=1,NB
AMEAN=AMT(I)+BAVG/UL*EVAC(I)*NSTART
BACKAR=AMEAN
COSTAR=ACUP*BACKAR
NRAR=0
BACKAR=BACKAR
COSTAL=COSTAR
NRAR=NRAR+1
BACKAR=BNAP(AMEAN,NRAR)
IF(BACKAR.LE.0.0)PAUSE "INC AREA * BACKAR=",#BACKAR 
COSTAR=ACUP*BACKAR+COMUP*NRAR
IF(COSTAR.LT.COST)GO TO 10
NR(I)=NRAR-1
BTOT=BTOT+BACKAR
NTOT=NTOT+NR(I)
IF(NSTART.EQ.1)GO TO 21
NSLOW=COMUP*NTOT+ACUP*BTOT
NSTART=1
GO TO 5
20
MM=0
NSHIGH=COMUP*NTOT+ACUP*BTOT
COST=ACUP*BTOT+COMUP*NTOT
COSTMN=COST
MMAX=INT((NSHIGH-NSLOW)/COMUP)
MM=MM+1
BAVG=BNAP(UTL,MM)
NTOT=0
BTOT=0.0
DO 50 I=1,NB
AMEAN=AMT(I)+BAVG/UL*EVAC(I)
NRAR=NR(I)
BACKAR=BNAP(AMEAN,NRAR)
IF(BACKAR.LE.0.0)PAUSE "INC NICP * BACKAR=",#BACKAR 
COSTAR=ACUP*BACKAR+COMUP*NRAR
IF(NRAR.EQ.1)GO TO 40
NRAR=NRAR-1
BACKAR=BACKAR
COSTAL=COSTAR
10
LISTING OF CODING CONTINUED

IF(NRAR.LT.0) GO TO 45
BACKAR=BNAP(CMEAN,NRAR)
COSTAR=ACUP*BACKAR+C0MUP*NRRAR
IF(COSTAR.LT.C0STAL) GO TO 40

45  N(R(I))=N(RAR)+1
NT0T=NT0T+NR(I)

50  BT0T=BT0T+BACKAL
COST=ACUP*BT0T+C0MUP*(NT0T+MM)
IF(COST.GE.C0STMN) GO TO 60
C0STMN=C0ST

NT0PT=NT0T
M0PT=MM
B0PT=BT0T
FOR I=1,NB: NR0PT(I)=NR(I)
MMAX=INT((COST-NSLOW)/C0MUP)
60  IF(MM.LT.MMAX) GO TO 30
FOR I=1,NB: NR(I)=NR0PT(I)
M=M0PT
NT0T=NT0PT
INV=MM+NT0T
BACK0=B0PT
W0PT=BACK0/AMDA
RETURN
END
REFERENCES

1. Rosenman, B. and Hoekstra, D. *A Management System for High Value Army Aviation Components*, AMC Inventory Research Office, Report No. TR 64-1, October 1964 (AD-452444)

DISTRIBUTION

Under-Secy of the Army, ATTN: Office of Operations Research
Asst Secy of Army (I&L)
Hqrs, U.S. Army Material Command
Commanding General
Deputy Commanding General
AMCRP
AMCQT
AMCDLS
AMCMO
AMCDT
AMCOR
AMCMS
AMCP-A-S
AMCQA
AMCRP
AMCRP-P
AMCRP-S
AMCRP-O
AMCRP-F
AMCRP-FR
AMCRP-FS
AMCRP-I
AMCRP-J
AMCRP-G
AMCRP-M
AMCRP-H
AMCRP-Q
AMCRP-D
AMCRP-K
AMCRP-V
AMCRP-W
AMCRP-N

Asst Dep Chief of Staff for Logistics, ATTN: ADCSLOG/LRAO
U.S. Army Pentagon 20310
ODCSLOG, Dir. of Supply, Room 1E558, Pentagon, Wash., D.C. 20310
Asst Dep Chief of Staff for Logistics (Plans, Doctrine & Systems)
U.S. Army, ATTN: LDSO, Wash., D.C. 20310
CO, Logistics Doctrine, Systems & Readiness Agency, New Cumberland
Army Depot, P.O. Box 2947, Harrisburg, Pa. 17105
Defense Logistics Studies Info Exchange ATTN: AMXMC-D
CG, USA Tank-Automotive Command
CG, USA Weapons Command
CG, USA Electronics Command
CG, USA Missile Command
CG, USA Missile Command, ATTN: AMSMI-QS
CG, USA Mobility Equipment Command
CG, USA Munitions Command
CG, USA Aviation Systems Command
CO, Army Automated Logistics Mgt Systems Agency, Box 14505, St. Louis, Mo. 63168 ATTN: AMKAL-SS
CO, Letterkenny Army Depot
ATTN: AMC Logistics Systems Support Center
ATTN: AMC Major Item Data Agency
CO, Lexington Army Depot, ATTN: Logistics Data Center
Commandant, US Army Logistics Mgt Ctr, ATTN: AMXMC-L
Fort Lee, Va.
Quartermaster General of the Marine Corps, Inventory Mgt. Ofc.
(CSR) Arlington Naval Cnd, Wash., D. C.
Defense Documentation Center
CG, US Air Force Logistics Command, WPAFB, Dayton, Ohio
ATTN: Mr. I. Katz, Opns Anal Off 35433
US Navy Fleet Materiel Support Office, Naval Supply Depot, Mechanicsburg, Pa. 17055
George Washington University, Log Rsch Proj, ATTN: Dr. S. Uaber, 707 22nd St., N.W., Wash., D. C.
Naval Postgraduate School, ATTN: Dept of Opns Anal, Monterey, Calif. 93940
Air Force Institute of Technology, ATTN: SLC, Bldg. 288, WPAFB, Dayton, Ohio 45433
The Army Library, Room 1A518, Pentagon, Wash., D. C. 20310
US Army Military Academy, West Point, N. Y.
Hqrs, US Army Materiel Cnd, ATTN: Dr. Badrig Kurkjian, AMCM-R
Logistics Management Institute, ATTN: Mr. Armand Weiss, 4701 Sangamore Road, Washington, D. C. 20016
Research Analysis Corp, ATTN: Mr. Lee Stoneback, McLean, Va. 22101
An Algorithm for Computing Optimum Stock Levels in a Two Level Maintenance System

Technical Report

W. Karl Kruse

January 1970

13

2

unlimited

AMC Materiel Command

An optimization algorithm for a multi-echelon model is given which does not assume convexity. The algorithm employs a bound technique to determine a finite search area. Updating the bound is used to increase the efficiency.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
</tbody>
</table>

### Security Classification

unclassified

HOLE
WT
RA
~r

---

Security Classification